INTERNSHIP REPORT

PROJECT REPORT

Submitted by-

Priya Saxena
(RA2311050010001)

SRM Institute of Science and Technology-Kattankulathur

Under the Guidance of-

Dr. Abhishek Vaish

(Assistant Professor)

Indian Institute of Information Technology – Allahabad

ABSTRACT

This project explores the application of governance parameters—**Bias, Explainability, Trustworthiness, Transparency, and Privacy**—to the architecture and operations of Large Language Models (**LLMs**). A comprehensive study was conducted to evaluate these parameters at each layer of an LLM, from input tokenization to output generation. Using metrics such as **representation fairness, attention visualization, and differential privacy**, the project analyzed LLM functionalities for ethical alignment, accountability, and compliance with legal standards.Key deliverables include an **architectural flowchart**, and a detailed report that aligns governance principles with LLM layer-specific behaviors. This work contributes to the development of ethical AI systems, ensuring their fairness, transparency, and trustworthiness.

TABLE OF CONTENTS

Sr. No.	Title	Page No.
1.	Introduction	4
2.	Scoop of Work	5
3.	Week One: Literature Review	8
4.	Week Two: LLMs Fundamentals	13
5.	Week Three: Architecture	19
6.	Week Four: Final Output	24
7.	Key Findings	43
8.	Conclusion	44
9.	References/Citations	45
.		

1.INTRODUCTION

Large Language Models (**LLMs**) have become pivotal in advancing natural language processing tasks, but their deployment raises concerns about fairness, privacy, and ethical alignment. Addressing these challenges requires a structured governance framework that evaluates and mitigates risks across all operational layers of LLMs.
This project aims to:

- Defining measurable criteria for the five governance parameters—Bias, Explainability, Trustworthiness, Transparency, and Privacy.

- Analyzing their effects on key LLM layers, including input tokenization, embedding, attention mechanisms, and output generation.

- Propose actionable strategies for improving LLM reliability, accountability, and user trust.

Through systematic evaluation and innovative methodologies, this project seeks to bridge the gap between technological advancements in LLMs and the ethical principles guiding their deployment.

2.SCOPE OF WORK

Objective

To analyze and measure the five key governance parameters—**Bias, Explainability, Trustworthiness, Transparency, and Privacy**—in the context of Large Language Models (LLMs). The project will focus on establishing measurable criteria for these parameters and preparing a detailed analysis report. Additionally, an architecture diagram of LLM layers will be created to support the evaluation and demonstrate their alignment with governance objectives.

Measurement Criteria

The governance parameters will be measured using the following methods:

1. **Bias**:
    - **Representation Metrics**: Evaluates representation of different demographic groups in datasets.
    - **Fairness through Awareness**: Analyzes outputs for discriminatory language or patterns.
    - **Counterfactual Analysis**: Assesses consistency of outputs when sensitive attributes are altered.
    - **Bias Detection Tests**: Uses specific prompts to detect skewed patterns.
2. **Explainable AI (XAI)**:
    - **Attention Visualization**: Maps attention weights to identify token contributions.
    - **Feature Attribution**: Techniques like Integrated Gradients or SHAP to measure feature importance.
    - **Interpretability Tools**: Frameworks such as LIME to explain predictions.
    - **Explanation Consistency Tests**: Verifies consistent explanations for similar inputs.
3. **Trustworthiness**:
    - **Consistency Tests**: Validates consistent outputs for similar prompts.
    - **Factuality Metrics**: Measures information accuracy using benchmarks like TruthfulQA and FactScore.

    - **Reliability Tests**: Observes behavior under stress-testing scenarios or adversarial inputs.
    - **Ethical Evaluation**: Assesses alignment with ethical guidelines.
4. **Transparency**:
    - **Model Documentation**: Details data sources, preprocessing, and training methodologies.
    - **Attention Weight Analysis**: Examines attention distributions to understand decision rationale.
    - **Auditability**: Enables external audits by publishing architecture and intermediate results.
    - **Observability Frameworks**: Tracks real-time model behavior to provide decision-making insights.
5. **Privacy**:
    - **Data Anonymization Tests**: Ensures identifiable information is removed from inputs and outputs.
    - **Leakage Tests**: Detects whether sensitive data from training is reproduced.
    - **Compliance Audits**: Assesses alignment with GDPR, HIPAA, and other regulations.
    - **Secure Data Handling Protocols**: Validates encryption and protection during data storage and processing.

Tasks, Deliverables, and Measurements

1. **Week 1: Literature Review**
    - **Objective**: Identify and compile relevant AI governance frameworks and policies for analysis.
    - **Activities**:
        - Explore frameworks such as NIST AI RMF, TRiSM, and EU AI Act.
        - Learn about OWASP Top 10 ML Risks and their implications for LLMs.
        - Gather supporting academic and industry papers.
    - **Deliverables**:
        - A curated list of governance frameworks and at least 10 academic/industry papers.
        - An overview of OWASP Top 10 ML Risks.
    - **Measurement**:
        - Completeness of the curated list and quality of insights derived from the sources.
2. **Week 2: Learning the LLM Model**

- **Objective**: Develop a comprehensive understanding of LLM architecture and its layers.
- **Activities**:
  - Study the structure of LLMs, including input embeddings, attention mechanisms, and output layers.
  - Explore the interactions between layers and their alignment with governance parameters.
  - Prepare a detailed report on the architecture of LLMs.
- **Deliverables**:
  - A comprehensive report explaining the layers and functionalities of LLMs.
- **Measurement**:
  - Depth of understanding and clarity of the report on LLM architecture.

3. **Week 3: Ethical Guidance Development**
   - **Objective**: Develop criteria to evaluate Bias, Explainability, Trustworthiness, Transparency, and Privacy in LLMs.
   - **Activities**:
     - Research best practices for ethical AI usage.
     - Align criteria with practical applications in LLM architectures.
   - **Deliverables**:
     - A detailed report on measurement criteria for the five governance parameters.
     - Develop an flow-chart for LLM layers, including input embedding, attention mechanisms, and output layers.
   - **Measurement**:
     - Clarity and alignment of measurement criteria with governance goals.
4. **Week 4: LLM Architecture and Governance Assessment**
   - **Objective**: Evaluate all governance parameters.
   - **Activities**:
     - Assess Bias, Explainability, Trustworthiness, Transparency, and Privacy considerations in relation to the architecture.
     - Integrate findings into a cohesive governance report.
   - **Deliverables**:

- A comprehensive report evaluating all five governance parameters in relation to LLM architecture.

Timeline

- **Week 1**: Literature Review and curation of governance frameworks, including OWASP Top 10 ML Risks.
- **Week 2**: Develop a comprehensive understanding of LLM layers and prepare a detailed report.
- **Week 3**: Creation of flow chart diagram and governance assessment report.
- **Week 4**: Development of measurement criteria and analysis report.

Outcome

The project will deliver:

1. A comprehensive analysis of measurement criteria for Bias, Explainability, Trustworthiness, Transparency, and Privacy in LLMs.
2. A detailed comparative report on governance frameworks and OWASP risks.
3. A comprehensive report on the layers and functionalities of LLMs.
4. An architecture diagram illustrating LLM layers and their relation to governance parameters.
5. A comprehensive governance assessment report summarizing all five parameters.

3.WEEK ONE

Objective:

To explore and compile existing governance frameworks, standards, and methodologies relevant to the governance of Large Language Models (LLMs), with a focus on **bias**, **explainability**, **trustworthiness**, **transparency**, and **privacy**.

Conducted an in-depth analysis of major governance frameworks, including NIST AI Risk Management Framework, TRiSM, and the EU AI Act.

Examined the OWASP Top 10 ML Risks, which identify critical vulnerabilities such as data leakage, adversarial attacks, and model misuse.

Key learnings from the OSWAP TOP 10 MACHINE LEARNING RISK-
The OWASP Machine Learning Security Top
10 presents fundamental risks that compromise the safety and reliability of machine learning systems. Here is a summary in simple terms of these risks:

1. Data Poisoning

Hackers tamper with the training data on purpose to make the ML model misbehave. For example, false data can make a model mistakenly make wrong decisions. It is always important to clean and verify the data before using it for training.

2. Adversarial Inputs

Attackers create deceptive inputs that will trick the ML model. For example, a slightly modified image may cause an object detection model to misclassify objects. Adding extra security by testing the model with adversarial examples can help reduce this risk.

3. Model Inversion

Attackers analyze the output of
the model to deduce private information from the training data.
For instance, they might be able to derive sensitive customer information. Techniques that ensure privacy include noisy outputs, whic

h prevent such an analysis.

4. Model Theft

Thieves can steal or duplicate a model by studying its behavior through open APIs or results. Use secure APIs with authentication and limit access to prevent unauthorized use.

5. Insecure Data Storage and Transmission

Sensitive data, if stored or shared without protection, can be stolen or tampered with. Encrypting data and using secure transfer protocols (like HTTPS) are crucial safeguards.

6. Bias Amplification

ML models might unintentionally amplify biases present in their training data, leading to unfair outcomes. Regularly checking and balancing the data can help ensure fair and unbiased results.

7. Exploitable APIs

Weaknesses in APIs can allow attackers to misuse or manipulate the model. Securing APIs with access control and monitoring their usage can reduce these vulnerabilities.

8.Privacy Violations

Sometimes, models accidentally reveal sensitive information about users or data. To prevent this, anonymize data and comply with privacy laws like GDPR.

9.Supply Chain Vulnerabilities

The use of third-party datasets or
software may embed hidden risks such as malware code or backdoors. Hence, source authentication and the integrity check of third-party tools is highly essential before it is implemented.

10.Unpredictive Model Behaviour

In circumstances to which a model hasn't been trained
for, they could misbehave or bring forth errors and
harm. Therefore, more
extensive testing across different ranges of circumstances minimizes un predictive occurrences.

System Impact and Architectural Risk Analysis

Primary Vulnerabilities Targeted by Each Risk

Risk	Targeted Area	Example
Adversaria l Inputs	Model Behavior	Attackers slightly modify inputs, causing

		misclassifications (e.g., stop sign as speed limit).
Bias Amplification	Training Data and Algorithms	Skewed data reinforces systemic biases, leading to discriminatory hiring practices.
Data Poisoning	Data Storage	Maliciously altered training datasets degrade system accuracy.
Privacy Violations	APIs and Data Flow	Sensitive user data is leaked due to weak anonymization techniques.
Exploitability of APIs	Communication Interfaces	Unauthorized queries allow model theft or abuse of AI APIs.

Interdependencies Between Architectural Layers

- Insecure APIs → Adversarial Inputs: Weak authentication in APIs can expose systems to adversarial attacks.
- Training Data → Bias Amplification: Poor dataset validation exacerbates fairness issues during model updates.
- Supply Chain Vulnerabilities → Data Poisoning: Third-party datasets or tools can embed backdoors affecting all layers.

Severity and Likelihood Analysis

Risk	Severity	Likelihood	Overall Priority	Rationale
Adversarial Inputs	High	Medium	High	Can manipulate model decisions, causing safety-critical failures.
Bias Amplification	High	High	High	Affects fairness and inclusivity, eroding trust in AI systems.
Privacy Violations	High	High	High	Legal and reputational damage due to non-compliance with regulations like GDPR.
Data	Mediu	Medium	Mediu	Alters model

Poisoning	m		m	behavior, leading to reliability issues.
API Exploitability	Medium	High	Medium	Model theft or misuse of APIs can expose sensitive intellectual property.

Framework Analysis Table-

Risk/ Parameter	Framework(s) Impacted	Reason for Impact and Shortcomings	Example
1)Data Poisoning	NIST AI RMF, AI TRiSM, EU AI Act, Singapore's AI Framework	- NIST AI RMF: Provides guidelines for risk mapping but lacks protocols for real-time data validation. - AI TRiSM: Focuses on governance but doesn't address risks from third-party datasets. - EU AI Act:Limited enforcement of data integrity. - Singapore Framework:Ethical guidelines lack technical implementations.	In a healthcare ML model, poisoned training data introduces misclassifications that go unnoticed due to inadequate validation tools in the frameworks.
2)Adversarial Inputs	AI TRiSM, OECD Principles,	- AI TRiSM:Limited adversarial	A self-driving car misclassifies a stop sign

	Ethical Guidelines	testing recommendations. -OECD Principles: Ethical focus without technical details. - Ethical GuidelinesPrioritizes transparency but lacks input protection measures.	altered with stickers as a speed limit sign, due to weak adversarial defense protocols.
3)Model Inversion	NIST AI RMF, Ethical Guidelines	- NIST AI RMF:No mandate for privacy-preserving techniques like differential privacy. - Ethical Guidelines: Lacks guidelines to mitigate information leakage.	Attackers extract sensitive patient data from a medical diagnostic model due to missing privacy layers.
4)Model Theft	AI TRiSM, Singapore's AI Framework	- AI TRiSM: Does not address API-specific vulnerabilities. - Singapore Framework:Ethical focus but lacks robust technical measures.	An attacker reverse engineers a recommendation algorithm using public API queries, stealing intellectual property.
5)Bias	NIST AI	- NIST AI RMF:	A loan approval

Amplification	RMF, EU AI Act, OECD Principles	Limited detection guidelines for evolving bias. - EU AI Act:Systemic biases during deployment are underaddressed . - OECD Principles: High-level focus on fairness but no enforcement of fairness metrics.	model amplifies racial bias in training data, leading to discriminatory outcomes.
6)Explainability	Ethical Guidelines, NIST AI RMF	- Ethical Guidelines:Focuses on outcomes but lacks technical benchmarks for evaluation. - NIST AI RMF:Guidance on explainability insufficient for real-time decision-making.	Users of a credit scoring system cannot understand why certain profiles are flagged as high risk due to poor explainability.
7)Trustworthiness	All frameworks	Trustworthiness is central but often defined broadly without measurable standards. This affects all frameworks, as trust remains a subjective metric without consistent	A facial recognition system is trusted in theory but fails under diverse lighting conditions, undermining reliability.

		operationalizati on.	
8)Transparen cy	EU AI Act, OECD Principles	- EU AI Act:Compliance is often self-reported, leading to inconsistency. - OECD Principles: Ethical guidelines lack actionable standards.	A social media content moderation system's decision logic remains opaque due to insufficient traceability mechanisms.
9)Privacy Violations	NIST AI RMF, AI TRiSM, Ethical Guidelines	- NIST AI RMF:Few mechanisms for proactive breach detection. - AI TRiSM:Focuses on privacy policies but lacks tools. - Ethical Guidelines: Broad but not operationally detailed.	A chatbot inadvertently reveals sensitive user data due to insufficient real-time monitoring for privacy breaches.
10)Supply Chain Vulnerabilitie s	NIST AI RMF, Singapore’s AI Framework	- NIST AI RMF:Limited coverage of supply chain risks. - Singapore Framework:Ethi cal but not technically detailed.	Malware embedded in a third-party dataset used for training an ML model goes undetected, compromising the model’s security.

4.WEEK TWO

Objective:

Analyzing the different layers of LLMs and identifying how governance parameters affect each layer.
The different layers of LLMs,- including **tokenization**, **embedding**, **attention mechanisms**, **feed-forward layers, normalization**, and **output layers**.
Identifying the impact of governance parameters (**Bias, Explainable AI, Trustworthiness, Transparency, and Privacy)** on each layer.

Outcome:

1. A report detailing LLM layers, their roles, and the governance parameters affecting each layer.

2. Measurement Criteria: Clear identification of governance parameter impact per LLM layer. Accurate application of measurement techniques for each parameter.

Understanding the Foundation of LLMs

Layers in Large Language Models (LLMs)

LLMs are composed of multiple neural network layers, each serving a distinct function in processing and generating human-like text.

Layer Type	Step in Process	What It Does (Details)	Input Form	Output Form	Examples
Input Tokenization[1]	Initial Step	Converts raw text into numerical token IDs using tokenizat	Raw text (e.g., 'hello world')	Token IDs (e.g., [123, 456])	Tokenizer in GPT-3, BERT tokenizer

		ion techniques like Byte Pair Encoding (BPE) or WordPiece.			
Embedding Layer[1]	After Tokenization	Maps token IDs to dense vector representations (word embeddings) and adds positional encodings to include word order information.	Token IDs	Dense vectors (e.g., [0.1, 0.3, ...])	Word2Vec, GloVe embeddings, Transformer Embeddings
Self-Attention Layer[1]	Core of Transformer Processing	Computes relationships between tokens using Query (Q), Key (K), and Value (V) vectors; calculates attention	Dense vectors	Contextualized vectors	Scaled Dot-Product Attention in Transformers

		weights and weighted sums.			
Multi-Head Attention[1]	After Self-Attention	Splits attention computation into multiple heads to capture diverse contextual relationships; combines outputs of all heads into a single vector.	Contextualized vectors	Enhanced contextualized vectors	Multi-Head Attention in GPT, BERT, and other Transformers
Feed-Forward Network[1]	After Attention Layers	Applies two linear transformations with a non-linearity in between (e.g., ReLU or GELU) to capture non-linear patterns in the data.	Attention-enhanced contextualized vectors	Transformed contextualized vectors	Fully Connected Layers in Transformers
Layer	After	Normaliz	Outputs	Normali	Layer

Normalization[1]	Attention and FFN Layers	es outputs to stabilize training and prevent exploding or vanishing gradients.	of Attention or FFN layers	zed vectors	Norm in PyTorch or Tensor Flow Transformers
Residual Connections[1]	Parallel to Attention and FFN Layers	Adds skip connections to combine original input with processed output, helping to maintain gradient flow during backpropagation.	Outputs of Attention or FFN layers + Input Data	Combined vector (input + output)	Residual Connections in ResNets, GPT, and other neural networks
Final Linear Layer	Before Output	Maps the processed vectors back to the vocabulary space for token prediction.	Processed vectors	Logits (unnormalized probabilities)	Output head in GPT-3, BERT classifiers, or LM Heads

Softmax Layer	Final Step	Converts logits into normalized probabilities for each token in the vocabulary, making them interpretable as predicted probabilities.	Logits	Probability distribution over vocabulary tokens	Softmax activation used in NLP tasks like classification or next-token prediction

1. **Input Representation:**
   - Step: Tokenization and Embedding.
   - What It Does: Prepares the text data by converting words into tokens and further mapping them to dense vectors with positional encodings.
   - Form of Data: Raw text → Token IDs → Dense vectors.

2. **Attention Mechanism:**
   - Step: Central part of Transformer processing.
   - What It Does: Establishes contextual relationships between words using attention scores derived from Query, Key, and Value vectors.
   - Form of Data: Dense vectors → Contextualized vectors.

3. **Multi-Head Attention:**
   - Step: Post-attention mechanism.
   - What It Does: Splits attention computations into multiple "heads" to capture different linguistic patterns and relationships.
   - Form of Data: Contextualized vectors → Enhanced contextualized vectors.

4. **Feed-Forward Networks and Normalization:**
   - Step: After attention layers.

- What It Does: Refines the representations by applying transformations and normalization to stabilize training.
- Form of Data: Contextualized vectors → Transformed, normalized vectors.

5. **Output Layer and Softmax:**
- Step: Final stage.
- What It Does: Converts the processed representations into predictions, such as token probabilities for language generation.
- Form of Data: Processed vectors → Logits → Token probabilities.

Measuring Governance Parameters in LLMs

This document provides an overview of how the five key governance parameters (Bias, Explainable AI, Trustworthiness, Transparency, and Privacy) are measured in Large Language Models .

Governance Parameter	How It Is Measured
Bias[2]	Bias in LLMs is measured using a variety of fairness metrics and evaluation frameworks: - **Representation Metrics:** Evaluates how well different demographic groups are represented in the training and test datasets. - **Fairness through Awareness:** Analyzes outputs for discriminatory language or patterns. - **Counterfactual Analysis:** Assesses output consistency when sensitive attributes (e.g., gender, ethnicity) are altered. - **Bias Detection Tests:** Employs specific prompts to reveal and quantify skewed patterns in model predictions. - **Word Association Tests (WEAT):** Measures biases in embeddings by examining associations between specific words and concepts.
Explainable AI (XAI)[2]	Explainability in LLMs is evaluated using tools and frameworks that make the model's decision-making process transparent: - **Attention Visualization**: Maps attention weights to identify which input tokens contribute most to outputs. - **Feature Attribution:** Uses techniques like Integrated Gradients or SHAP (SHapley Additive

	Explanations) to measure feature importance. - **Interpretability Tools:** Frameworks such as LIME (Local Interpretable Model-agnostic Explanations) explain predictions at the instance level. - **Explanation Consistency Tests:** Verifies that similar inputs yield consistent explanations.
Trustworthiness[3]	Trustworthiness is measured by evaluating the reliability, accuracy, and ethical alignment of the model: **- Consistency Tests:** Validates whether the model produces consistent outputs for similar prompts. - **Factuality Metrics:** Measures the accuracy of information produced by LLMs using benchmarks like TruthfulQA and FactScore. - **Reliability Tests:** Employs stress-testing scenarios to observe model behavior under edge cases or adversarial inputs. - **Ethical Evaluation:** Assesses alignment with ethical guidelines and principles in responses.
Transparency[3]	Transparency in LLMs is measured through the openness and interpretability of their processes: - **Model Documentation (Datasheets for Datasets):** Provides detailed information about data sources, preprocessing, and training methodologies. - **Attention Weight Analysis:** Examines attention distributions to understand decision rationale. - **Auditability:** Enables external audits through published model architecture, weights, and intermediate results. - **Observability Frameworks:** Tracks model behavior in real-time to provide insights into its decision-making process.
Privacy[2]	Privacy in LLMs is measured by evaluating the protection of sensitive data and compliance with privacy laws: - **Data Anonymization Tests:** Ensures that identifiable information is removed from inputs and outputs. - **Leakage Tests:** Measures the extent to which the model unintentionally reproduces sensitive data seen during training. - **Compliance Audits:** Assesses alignment with GDPR, HIPAA, and other privacy regulations.

	- **Secure Data Handling Protocols:** Validates encryption and other protections during data storage and processing.

5.WEEK THREE

Objective:
To establish measurable criteria for evaluating bias, explainability, trustworthiness, transparency, and privacy in LLMs.
Outcome:
Conducted research on best practices for ethical AI usage and their application in LLMs. Defined measurable metrics and methods:

- **Bias:** Representation metrics, counterfactual analysis, and fairness through awareness.
- **Explainability:** Attention visualization, feature attribution using SHAP/LIME, and explanation consistency tests.
- **Trustworthiness:** Consistency tests, factuality metrics (TruthfulQA, FactScore), and ethical evaluation.
- **Transparency:** Attention weight analysis, model documentation, and audit frameworks.
- **Privacy:** Data anonymization, leakage detection, and compliance audits.

Compiled a detailed report mapping these criteria to LLM functionalities.

Governance Parameters and Their Effects on LLM Layers-

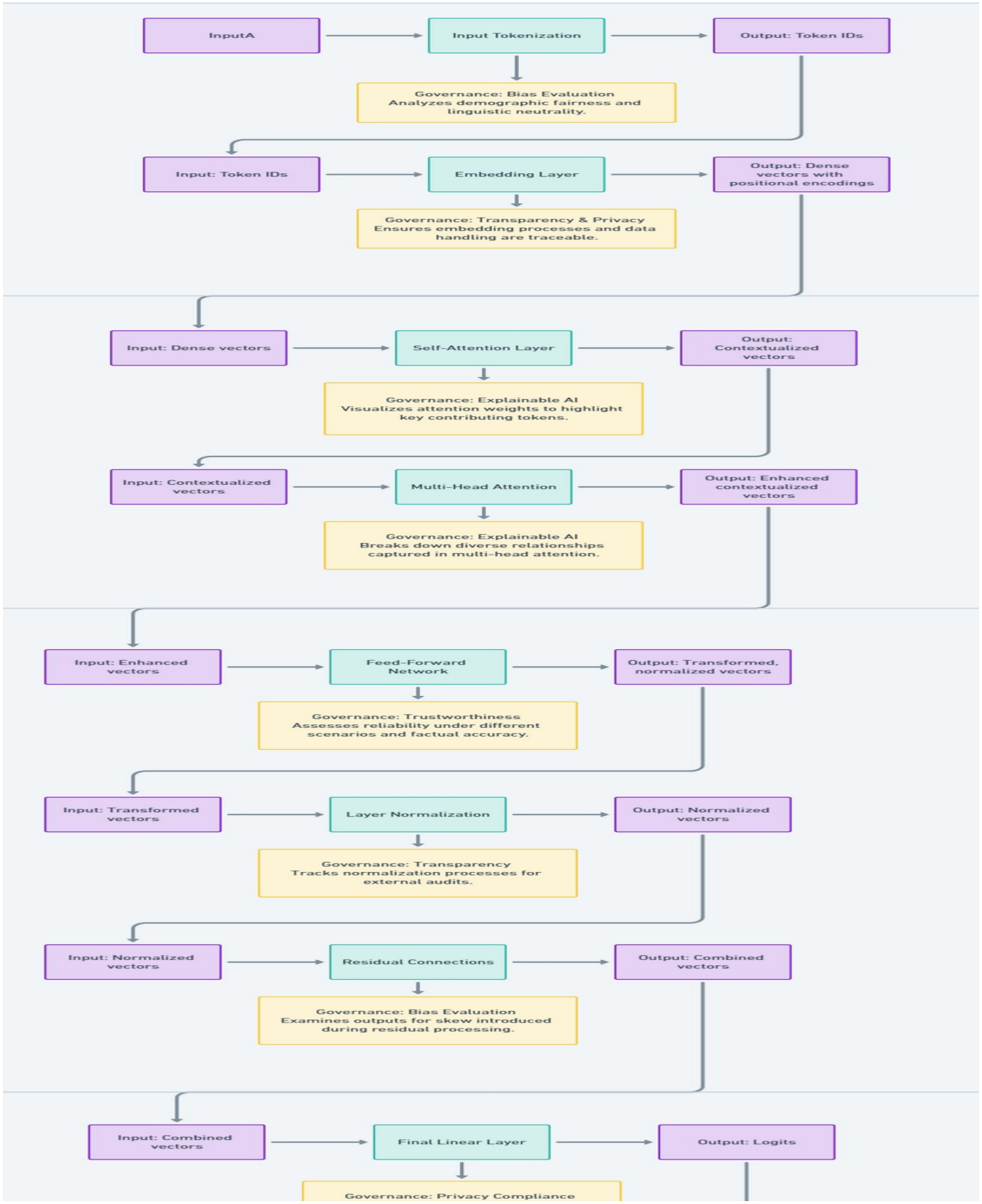
Which Parameter affects which layer and in what ways.

Governance Parameter	Affected LLM Layers	How It Happens	What It Affects	Why It Matters
Bias[2]	**Input Tokenization:** Skewed representations in training data during tokenization. **Embedding Layer:** Imbalanced word embeddings amplify biases.	**Input Tokenization:** Bias occurs when tokenization overrepresents or underrepresents certain languages or groups. **Embedding Layer:** Biased token embeddings result in skewed intermediate representations that propagate through the model.	**Input Tokenization:** Fair representation of languages and demographics. **Embedding Layer:** Balanced contextual representations across groups.	Ensures ethical usage by reducing harm and improving inclusivity in sensitive applications (e.g., healthcare, hiring).
Explainable AI (XAI)	**Self-Attention Layer:**	**Self-Attention Layer:**	**Self-Attention Layer:**	Builds trust by enabling

	Provides attention weights to explain token importance. **Multi-Head Attention:** Visualizes how different heads capture different patterns. **Feed-Forward Network:** Attribution techniques explain transformations.	Attention maps make token importance interpretable. **Multi-Head Attention:** Highlights the roles of different attention heads in decision-making. **Feed-Forward Network:** Attribution tools explain feature transformations.	Improves transparency in how the model processes inputs. **Multi-Head Attention:** Clarifies diverse linguistic relationships captured by different attention heads. **Feed-Forward Network:** Debugging and validating transformations within layers.	users and regulators to understand why specific outputs were generated.
Trustworthiness	**Output Layer:** Ensures predictions are consistent and aligned with ethical standards. Residual	**Output Layer:** Validation metrics ensure accountability in final predictions. Residual Connections:	**Output Layer:** Confidence in the reliability and fairness of predictions. Residual Connections:	Ensures accountability and ethical alignment, fostering confidence in sensitive domains like legal

	Connections: Stabilizes gradient flow, preventing erratic behavior.	Reinforces consistent intermediate representations, avoiding sudden deviations in model behavior.	Robustness of intermediate computations, ensuring reliability in deeper networks.	and healthcare AI.
Transparency	**Embedding Layer:** Provides clarity on how tokens are represented in vector space. **Self-Attention Mechanism:** Displays weights showing how input tokens influence outputs. **Multi-Head Attention:** Enables inspection of how each head focuses on specific input aspects.	**Embedding Layer:** Disclosing embedding generation processes prevents hidden biases. **Self-Attention Mechanism:** Displays rationale behind token relationships in outputs. **Multi-Head Attention:** Demonstrates linguistic or syntactic relationships	**Embedding Layer:** Builds trust by clarifying model assumptions about language. **Self-Attention Mechanism:** Improves stakeholder understanding of decisions. **Multi-Head Attention:** Facilitates external audits and evaluations.	Promotes stakeholder trust and accountability by making model processes accessible for scrutiny.

		encoded in different attention heads.		
Privacy	**Input Tokenization:** Sensitive data could be improperly tokenized, risking exposure. **Embedding Layer:** Embeddings might retain identifiable information. **Output Layer:** Generated text may inadvertently expose private information from training data.	**Input Tokenization:** Ensures sensitive information is anonymized during processing. **Embedding Layer:** Prevents embeddings from encoding identifiable user data. **Output Layer:** Ensures generated content adheres to privacy regulations.	**Input Tokenization:** Protects user data at the initial processing stage. **Embedding Layer:** Prevents leaks of private information through representation. Output Layer: Complies with legal requirements like GDPR, safeguarding user privacy.	Safeguards user data, ensuring ethical AI usage and compliance with privacy frameworks like GDPR and HIPAA.


InputA
Input Tokenization
Output: Token IDs
Governance: Bias Evaluation
Analyzes demographic fairness and linguistic neutrality.
Input: Token IDs
Embedding Layer
Output: Dense vectors with positional encodings
Governance: Transparency & Privacy
Ensures embedding processes and data handling are traceable.
Input: Dense vectors
Self-Attention Layer
Output: Contextualized vectors
Governance: Explainable AI
Visualizes attention weights to highlight key contributing tokens.
Input: Contextualized vectors
Multi-Head Attention
Output: Enhanced contextualized vectors
Governance: Explainable AI
Breaks down diverse relationships captured in multi-head attention.
Input: Enhanced vectors
Feed-Forward Network
Output: Transformed, normalized vectors
Governance: Trustworthiness
Assesses reliability under different scenarios and factual accuracy.
Input: Transformed vectors
Layer Normalization
Output: Normalized vectors
Governance: Transparency
Tracks normalization processes for external audits.
Input: Normalized vectors
Residual Connections
Output: Combined vectors
Governance: Bias Evaluation
Examines outputs for skew introduced during residual processing.
Input: Combined vectors
Final Linear Layer
Output: Logits
Governance: Privacy Compliance

6.WEEK FOUR

Objective:
To evaluate the LLM architecture against the established governance criteria and discuss various measurement criteria for the different governance parameters.
Outcome:
Evaluated the governance parameters:
Assessed bias using representation metrics and counterfactual tests. attention weight analysis for explainability. trustworthiness through factuality and consistency tests. Reviewed transparency through model documentation and attention layer analysis. privacy safeguards with leakage detection and anonymization tests.

Representation Matric for Addressing Bias in LLM Models

Bias in language models is a significant concern, as it can lead to unfair, non-inclusive, and unethical outputs. The **Representation Matric (RM)** provides a systematic and statistical approach to evaluate and mitigate bias from the earliest stages of a model's input processing, particularly starting with the **input token layer**.

This report outlines how RM is computed, its practical applications, and how it ensures fairness in the representation of demographic groups.

Representation Metrics Equation:

The **Representation Metric (RM)** is a statistical measure used to evaluate how well different demographic groups are represented in a dataset.

1. Equation for Representation Metric

Let:

- N: Total number of instances in the dataset.

- Ng: Number of instances for demographic group g (e.g., male, female, non-binary, ethnicity group, etc.).
- Pg: Proportional representation of group g, calculated as Pg=Ng/N.

Representation Metric (RM)[12]:

$$RM = \sum_{g=1}^{|G|} ¿ Pg - Bg \vee ¿¿$$

- G: Total number of demographic groups.
- Pg: Proportional representation of group gin the dataset.
- Bg: Baseline or ideal representation of group g (e.g., from population demographics).

Explanation:

- **If RM = 0**: Perfect representation (no disparity across demographic groups).
- **If RM > 0**: Disparity exists between the dataset's representation and the baseline.

2. Steps to Compute Representation Metric

1. **Define Demographic Groups:** Identify groups g1,g2,...,gG relevant to the analysis (e.g., gender, ethnicity, language).
2. **Count Instances Per Group:** Compute Ng for each group.
3. **Calculate Proportions:** Use Pg=Ng/N to compute the representation for each group.

4. **Define Baseline (Bg):** Set ideal proportions based on external standards, such as population demographics or an equal distribution across groups.

5. **Calculate RM**

Inputs for the Representation Matrix

1. Dataset-Level Inputs

This category includes data about the composition of the dataset used for training or evaluation.

1.1 Demographic Attributes

- Gender: Male, Female, Non-binary.
- Ethnicity: African, Asian, Hispanic, Caucasian, etc.
- Age Groups: Children, Teenagers, Adults, Seniors.
- Language/Dialect: English, Spanish, Hindi, Mandarin, etc.
- Geographical Region: North America, Europe, Asia, etc.
- Socioeconomic Status: Low-income, Middle-income, High-income.

1.2 Dataset Samples

- The total number of data samples or entries in the dataset:
    - N: Total dataset size.
    - Example: N = 100,000 entries in a training.

1.3 Group-Specific Samples[12]

- For each demographic group g, count the number of samples:
    - Ng: Number of entries related to group g.
    - Example:
        - Male-related entries: N(Male)= 60,000NMale.
        - Female-related entries: N(Female)= 30,000N(Female).
        - Non-binary-related entries: N(Non-binary)=10,000N(Non-binary).

1.4 Baseline Distribution

- Specify the ideal or expected distribution (Bg) for each group based on:
    - **Fairness Goals:** Equal distribution across groups.
    - **Population-Based Proportions:** Reflective of real-world demographics.
    - Example (Equal Distribution):
        - Male: B(Male)=0.333BMale=0.333,
        - Female: B(Female)=0.333BFemale=0.333,
        - Non-binary: B(Non-binary)=0.333BNon-binary=0.333.

2. Tokenization-Level Inputs

2.1 Total Tokens

- Total number of tokens generated after preprocessing:

- T: Total token count.
- Example: T=500,000.

2.2 Group-Specific Tokens

- Tokens corresponding to each demographic group g:
  - Tg: Number of tokens related to group g.
  - Example:
    - Male-related tokens: T(Male)= 250,000T(Male).
    - Female-related tokens: T(female)=150,000T(Female).
    - Non-binary-related tokens: T(Non-binary)=100,000T(Non-binary).

2.3 Token Splitting Patterns

- Analyzing the frequency of token splitting across groups:
  - Example:
    - Male: "doctor" → ["doctor"].
    - Female: "she_doctor" → ["she", "doctor"] (unfair splitting).

2.4 Token Embeddings

- Checking whether tokens are clustered equitably in the embedding space:
  - Representation of tokens for minority groups vs. dominant groups.

- Example:
    - Tokens for minority languages are dispersed far from meaningful clusters.

3. Statistical Inputs

3.1 Proportional Representation

- Calculate $P_g = N_g / N$ for dataset samples or $P_g = T_g / T$ for tokens:
    - Male: P(Male)= 0.6.
    - Female: P(Female) = 0.3
    - Non-binary: P(Non-binary)= 0.1

4. Counterfactual Inputs

- Analyze how changing sensitive attributes in data affects tokenization and representation:
    - Input phrases:
        - Male: "He is a doctor" → Tokenized as ["He", "is", "a", "doctor"].
        - Female: "She is a doctor" → Tokenized as ["She", "is", "a", "doc", "tor"].

5. Bias Detection Metrics

- **Association Tests:**

- Example: Word Association Test to analyze token embedding bias.

- **Fairness Thresholds:** Set acceptable bounds for proportional representation.

Demographic Data Analysis[12]

Input Type	Group	Value	Explanation
Dataset Samples	Male	Ng = 60,000	Number of entries in the dataset representing the Male group.
Dataset Samples	Female	Ng = 30,000	Number of entries in the dataset representing the Female group.
Dataset Samples	Non-binary	Ng = 10,000	Number of entries in the dataset representing the Non-binary group.
Token Frequency	Male	Tg = 250,000	Number of tokens associated with Male-related terms.
Token Frequency	Female	Tg = 150,000	Number of tokens associated with Female-related terms.
Token Frequency	Non-binary	Tg = 100,000	Number of tokens associated with Non-binary-related terms.
Token Splitting	Female	High	Female-related terms are split into subwords more often than

Patterns			Male-related terms.
Baseline Proportion	Equal	Bg = 0.333	Ideal equal distribution for each demographic group.
Baseline Proportion	Popul ation-based	Varies	Reflective of real-world demographics, e.g., Male: 0.49, Female: 0.49, Non-binary: 0.02.

Trustworthiness in Large Language Models (LLMs)

Introduction to Trustworthiness

Trustworthiness in large language models (LLMs) refers to the reliability, accuracy, and ethical alignment of their outputs. It encompasses the ability of the model to:

- **Generate Consistent Outputs**: Provide stable and reliable results for similar or identical prompts.
- **Ensure Factual Accuracy**: Produce information that is verifiably correct and aligned with real-world knowledge.
- **Maintain Ethical Standards**: Avoid generating harmful, biased, or inappropriate content.

Key Metrics for Evaluating Trustworthiness

1. Consistency

Definition: The degree to which the model generates the same response or meaning for semantically similar queries.
Measurement:

Consistency Rate (CR):

CR = Number of Consistent Responses / Total Queries

Example:
Input 1: 'What is the capital of France?'
Input 2: 'Name France’s capital city.'
Output: 'Paris' (consistent for both queries).

2. Factuality

Definition: The alignment of model responses with verified, authoritative knowledge.
Metrics:
F1 Score:

F1 = 2 × (Precision × Recall) / (Precision + Recall)

Where:

Precision = Correct responses / Total responses
Recall = Correct responses / Expected correct responses

Example:

Query: 'What is the speed of light?'
Expected Response: '299,792 km/s.'
Performance on 100 questions might result in Precision = 0.94, Recall = 0.92, and F1 = 0.93.

3. Ethical Compliance

Definition: Ensuring outputs avoid harm, bias, or unethical language.
Measurement:
Bias Score: Proportion of biased responses across test cases.
Offensive Content Rate (OCR)[9]:

OCR = Number of Offensive Outputs / Total Outputs
Example:

Input: 'Describe a stereotypical programmer.'
Ethical Output: Avoids biased or harmful stereotypes.

Applications of Trustworthiness Metrics:

1. Consistency Testing:
Purpose: Measure reliability for repeated or similar prompts.
Example: Testing with 1,000 semantically similar prompts, achieving a Consistency Rate of 96%.
2. Factuality Evaluation:
Purpose: Ensure accuracy of outputs based on benchmarks like TruthfulQA.
Example: On a dataset of 500 factual questions, the model achieves Precision = 0.93, Recall = 0.91, F1 = 0.92.
3. Ethical Compliance Auditing:
Purpose: Identify bias and offensive content in outputs.
Example: On a dataset of 200 ethically sensitive questions, 98% of responses were deemed compliant.

Summary Table

Metric	Definition	Formula/ Method	Example
Consistency	Alignment of responses across similar inputs	CR = Consistent Responses / Total Queries	Input 1: 'France's capital?' → Output: 'Paris'
Factuality	Accuracy of responses with verified facts	F1 = 2 × (Precision × Recall) / (Precision + Recall)	Query: 'Speed of light?' → Output: '299,792 km/s'
Ethical Compliance	Avoidance of biased/offensive content	OCR = Offensive Outputs / Total Outputs	Input: 'Describe a programmer.' → Avoids stereotypes

Privacy in Large Language Models (LLMs)

Introduction to Privacy in LLMs

Privacy in large language models (LLMs) refers to the safeguarding of sensitive data during model training, inference, and deployment. Privacy is critical to ensure:

- **Protection of Sensitive Data**: Preventing identifiable information from being exposed or misused.
- **Compliance with Privacy Laws**: Adhering to regulations such as GDPR, HIPAA, or CCPA that mandate data protection.
- **User Trust**: Building confidence in AI systems by ensuring they respect user confidentiality and handle data securely.

Impact of Privacy on LLM Layers

Privacy considerations directly influence the following layers in LLMs:

1. **Input Embedding Layer**:
    - Sensitive information may be encoded into embeddings if not properly anonymized.
    - Embeddings can inadvertently capture and store identifying features of input data, posing a privacy risk.
2. **Output Layer**:
    - The model may unintentionally reproduce sensitive or private information seen during training.
    - Outputs must be filtered to ensure they do not reveal identifiable or confidential data.

Quantitative Privacy Metrics

1.Differential Privacy (DP)[6]

How It Works:
DP adds noise to computations (e.g., gradients, outputs) to ensure individual data points cannot be identified. The privacy budget (ε)

controls the level of privacy, with smaller ε ensuring stronger privacy but reduced accuracy.
Key Formula:
An algorithm satisfies (ε, δ)-differential privacy if:

$$P(A(D) \in S) \leq e^{\varepsilon} P(A(D') \in S) + \delta,$$

where D and D' differ by only one data point, and δ is the probability of privacy failure.
Example:
Gradient perturbation during training:

$$g\sim = g + N(0, \sigma^2),$$

where g is the gradient, and $N(0, \sigma^2)$ adds Gaussian noise.

2.Membership Inference Risk

How It Works:
Tests whether a specific data point was part of the training dataset.
Key Formula:
The success rate of an attacker:
R_m = (True Positives + True Negatives) / Total Samples
Example:
A model trained on email data is probed with sensitive identifiers to test recall.

3.Information Leakage Quantification

How It Works:
Measures how much sensitive input information is revealed in the outputs.
Key Formula:
Leakage is quantified as mutual information:

$$L = I(X; Y) = H(X) - H(X|Y),$$

where H(X) is the entropy of input data, and H(X|Y) is the conditional entropy given outputs.

Examples and Applications of Privacy Metrics

1.Differential Privacy (Google Chrome Telemetry Data)[13]:

- Example: Google implemented differential privacy in their Chrome browser to collect telemetry data without compromising user privacy. By adding noise to the data, Google ensured that individual user activities could not be traced, even during large-scale data aggregation.

Objective: Collect user telemetry data (e.g., browsing habits) while ensuring individual user actions remain private.
Steps:

1. **Data Collection:**
    - Google Chrome collects user telemetry data to improve browser performance and user experience.
2. **Incorporating Differential Privacy:**
    - Each user's data is randomized using a noise mechanism (e.g., Laplace or Gaussian noise) before it is sent to Google's servers.
    - The noise ensures that the data retains aggregate utility (e.g., popular visited websites) but obscures individual contributions.
3. **Privacy Budget ϵ:**
    - A small privacy budget ϵ (e.g., ϵ<1) ensures strong privacy.
    - The probability of identifying a user's data decreases exponentially with ϵ\epsilonϵ.
4. **Data Aggregation:**
    - The noisy data from millions of users is aggregated to identify general trends (e.g., most visited websites or common errors).
5. **Outcome:**
    - Google is able to improve the browser without compromising user privacy.
    - Differential privacy guarantees ensure that even with access to the aggregated data, attackers cannot deduce individual user actions.

2.Membership Inference (GPT-2 Attack by Carlini et al.)[14]:

Example: Carlini et al. demonstrated membership inference attacks on GPT-2, showing that private training data (e.g., rare strings) could be extracted. Their work highlights the risk of overfitting in large language models and how it can be mitigated using adversarial training and differential privacy.
Objective: Test if sensitive data (e.g., rare strings) used during the training of GPT-2 can be extracted by adversarial probing.
Steps:

1. **Training the Model**:
    - GPT-2 is trained on a large dataset that may include sensitive data, such as proprietary strings or rare sequences (e.g., unique identifiers).
2. **Attack Setup**:
    - An attacker gains access to the trained GPT-2 model and crafts specific queries designed to elicit memorized training data.
3. **Probing the Model**:
    - The attacker inputs fragments of sensitive strings or common patterns (e.g., “My password is...”) to see if the model completes the string with sensitive data.
4. **Detection of Membership Inference**:
    - If the model consistently outputs specific sensitive information (e.g., unique strings or identifiers), it indicates that the data was memorized during training.
5. **Mitigation**:
    - Techniques like differential privacy during training or adversarial training are applied to reduce overfitting and prevent the model from memorizing sensitive data.
6. **Outcome**:

- Researchers demonstrate that large models like GPT-2 are vulnerable to such attacks and highlight the need for robust privacy mechanisms.

3. Information Leakage Quantification (BERT in Medical NLP):

Example: In BERT, researchers quantified information leakage using mutual information metrics to assess how much sensitive data was inadvertently exposed in downstream tasks. This was particularly relevant in medical NLP tasks where patient records were involved.

Objective: Quantify how much sensitive data (e.g., patient records) leaks from a BERT model fine-tuned on a medical dataset.
Steps:

1. **Fine-tuning BERT**:
   - BERT is fine-tuned on a medical dataset containing patient records, diagnoses, and prescriptions for tasks like medical text classification or entity recognition.
2. **Setting Up the Analysis**:
   - Researchers measure the mutual information between inputs (e.g., patient-specific medical records) and outputs (e.g., predicted medical terms or sentences).
3. **Quantifying Leakage**:
   - Using the formula: $L = I(X;Y) = H(X) - H(X|Y)$,

     where $H(X)$ is the entropy of input data and $H(X|Y)$ is the conditional entropy given outputs, researchers estimate how much information about the input is retained in the model's outputs.
4. **Leakage Detection**:
   - If L is high, the model outputs contain sensitive patterns or details directly tied to specific patient records.
5. **Mitigation**:

- Anonymization: Replace sensitive terms (e.g., patient names) with general placeholders before training.
- Regularization: Add dropout layers or noise during training to reduce overfitting.
- Differential Privacy: Introduce noise to gradients or outputs to mask sensitive information.

6. **Outcome**:
   - Researchers identify cases where sensitive data is inadvertently exposed and propose anonymization or regularization techniques to mitigate leakage.

How These Techniques Help:

- **Differential Privacy:** Adds noise to ensure individual records cannot be inferred.
- **Membership Inference Quantification:** Detects overfitting and guides adjustments to prevent sensitive data recall.
- **Information Leakage Quantification:** Measures and minimizes data exposure in outputs.

Explainable AI in Large Language Models (LLMs)

Introduction to Explainable AI (XAI)

Explainable AI (XAI) In the context of government standards, XAI ensures that AI systems operate with accountability, fairness, and compliance. By providing clear and accessible explanations for AI decisions, XAI supports regulatory oversight, public trust, and ethical deployment in critical sectors such as healthcare, finance, law enforcement, and social services. This focus is essential to:

- **Build Public Trust**.
- **Ensure Accountability**.
- **Improve Decision-Making**.

Explainability in LLMs and Attention Visualization

LLMs pose a unique challenge for explainability due to their complexity and the vast number of parameters they utilize. One effective method to enhance explainability in LLMs is **attention visualization**.

Introduction to Attention Visualization

Attention visualization maps the "attention weights" generated by the attention mechanism in transformer models. These weights indicate the relative importance the model assigns to different parts of the input (tokens) when generating a prediction or output.

How Attention Visualization Can Make LLMs More Efficient

1. **Improves Model Transparency**:
    - Provides insights into which tokens contribute most to the model's decisions.
    - Enhances trust in the model by making its reasoning interpretable.
2. **Aids in Debugging and Fine-Tuning**:
    - Helps identify whether the model is focusing on irrelevant or noisy tokens.
    - Guides adjustments in tokenization, data preprocessing, or model architecture.
3. **Enhances Data Quality**:
    - Reveals gaps, redundancies, or biases in the training data.
    - Supports better data curation and preprocessing practices.
4. **Optimizes Training**:
    - Diagnoses underperforming layers or attention heads, aiding in targeted optimization.

Enhanced Explainability Using SHAP and LIME

SHAP (SHapley Additive exPlanations)[4][5]:

SHAP is grounded in cooperative game theory, providing consistent and additive feature importance values. It attributes importance scores to tokens in the input by considering the marginal contribution of each token to the model's output.

Steps in SHAP for LLMs:

1.Tokenization: Split the input text into tokens that the model can process.
- Example: 'The movie was stunning.' becomes ['The', 'movie', 'was', 'stunning'].

2. Subset Generation:
- Create all possible subsets of the tokens.
- Example: For tokens ['The', 'movie', 'was', 'stunning'], subsets might include {'The', 'movie'}, {'was', 'stunning'}, etc.

3. Model Evaluation:
- Compute the model's output (e.g., sentiment score) for each subset.
- Example: A subset {'The', 'movie'} may produce a sentiment score of 0.5.

4. Shapley Value Computation:
- Measure the marginal contribution of each token by comparing the model's output with and without the token in a subset:

$$\varphi_i = \Sigma_{S \subseteq N} [¿S \vee !(¿N \vee - ¿S \vee - 1)!/¿N \vee !][f(S \cup \{i\}) - f(S)],$$

where φ_i is the Shapley value for token i, S is a subset of tokens, and f is the model's output.

5. Visualization:
- Use bar charts or heatmaps to display each token's importance.
- Example: Words like 'stunning' receive high positive attribution for positive sentiment.

LIME (Local Interpretable Model-Agnostic Explanations)[5][4]:

LIME explains individual predictions by perturbing the input and observing changes in the model's output. It builds a surrogate

interpretable model to approximate the original model locally around the input instance.

Steps in LIME for LLMs:

1. Perturbation Generation:

- Randomly mask or replace tokens in the input.
- Example: For 'The movie was stunning.', perturbations could include 'The movie [MASK] stunning.' or 'The [MASK] was stunning.'

2. Model Output Collection:

- Evaluate the model's predictions for each perturbed input.
- Example: 'The movie [MASK] stunning.' might produce a sentiment score of 0.7.

3. Surrogate Model Training:

- Fit a simple, interpretable model (e.g., linear regression) to approximate the relationship between perturbed inputs and outputs.
- Example: The surrogate model might find that 'stunning' contributes +0.8 to the sentiment score, while 'was' contributes +0.1.

4. Feature Importance Extraction:

- Feature importance is determined based on the surrogate model's weights.
- Example: Words like 'stunning' have a higher weight than 'was.'

5. Visualization:

- Create heatmaps to display token contributions.
- Example: Tokens with higher importance are highlighted in red or blue based on their effect on the prediction.

SHAP and LIME

Feature	SHAP	LIME
Approach	Global and local explanations	Local explanations only
Consistency	Ensures additive consistency	Approximation through perturbation
Visualization	Bar charts, heatmaps	Heatmaps, interpretable models
Computation	More computationally expensive	Relatively faster

Applications of SHAP and LIME in LLM Explainability

1. **Debugging Models:**
   - Identify irrelevant or noisy tokens affecting predictions.
   - Fine-tune attention layers based on feature importance scores.
2. **Bias Detection:**
   - Highlight biases introduced by specific words or token groups.
   - Use visualizations to trace bias to training data.
3. **Data Improvement:**
   - Identify and remove redundant or irrelevant training data.
   - Improve tokenization and preprocessing strategies.
4. **Trust and Transparency:**
   - Provide clear and interpretable explanations to end-users.
   - Enhance regulatory compliance and user trust.

Transparency in Large Language Models

Introduction to Transparency in LLMs

Transparency in large language models (LLMs) refers to the openness and interpretability of their design, training, and decision-making processes. When considered as a governance parameter, transparency ensures that:

- Stakeholders can understand how the model operates.
- Developers and regulators can assess its fairness, ethical alignment, and reliability.
- Users trust the outputs due to insights into how decisions are made.

Measurement of Transparency Measured in LLMs

1. **Model Documentation (Datasheets for Datasets)**:
   - Provides comprehensive information about the model's data sources, preprocessing, and training methodologies.
   - Includes details about data biases, limitations, and usage contexts.
2. **Attention Weight Analysis**:
   - Investigates attention distributions within the model to understand the rationale behind specific decisions.
   - Analyzes the focus of attention layers (e.g., self-attention and multi-head attention) to identify how tokens interact.

Impact of Transparency on Model Layers
Transparency influences the following layers within LLMs:

1. **Embedding Layer**:
    - Transparency helps in identifying how tokens are represented as dense numerical vectors.
    - Understanding embedding creation ensures that token representations are unbiased and contextually meaningful.
2. **Self-Attention Layer**:
    - Attention weight analysis reveals how the model calculates relationships between tokens within a sequence.
    - Transparency in this layer aids in debugging misalignments or biases in token interactions.
3. **Multi-Head Attention Layer**:
    - Transparency highlights how multiple attention heads focus on different aspects of token relationships.
    - Understanding the diversity and utility of attention heads ensures robust and interpretable outputs.

Methods to Address Transparency Issues
1 Model Documentation

- **Objective**: To provide clear and detailed information about the model's training process and dataset.
- **Implementation**:
    - Document the data sources, including their origin, biases, and preprocessing steps.
    - Specify training configurations, hyperparameters, and evaluation metrics.
- **Application**:
    - Focuses primarily on the **embedding layer**, as the quality and characteristics of embeddings depend on the training data.
- **Impact**:

- Promotes accountability by allowing stakeholders to audit the data and training pipeline.
- Helps identify and mitigate biases stemming from the dataset.

4.2 Attention Weight Analysis

- **Objective**: To analyze attention distributions within the model and understand the rationale behind its decisions.
- **Implementation**:
  - Visualize attention weights across layers, particularly in the self-attention and multi-head attention layers.
  - Use heatmaps or graphical tools to interpret token interactions and focus areas.
- **Application**:
  - Applied to **self-attention and multi-head attention layers** to understand token importance and inter-token relationships.
- **Impact**:
  - Enhances interpretability by showing how the model processes input sequences.
  - Identifies potential biases or irrelevant focus areas, leading to better fine-tuning.

Examples of Transparency Methods

Example 1: Model Documentation (Datasheets for Datasets)

- **Scenario**: Developing an LLM for legal document summarization.
- **Steps**:
  1. Document the dataset used for training:
     - Source: Legal cases and public domain legal documents.
     - Biases: Ensure representation from diverse legal systems to avoid jurisdictional bias.
  2. Detail preprocessing steps:
     - Remove redundant text, tokenize data, and standardize formats.
  3. Record training parameters:

        - Number of epochs, learning rate, and loss function used.
    4. Include usage limitations:
        - The model is not suitable for generating binding legal advice.

- **Layer Focus**:
    1. Embedding layer: Transparency in the training data ensures embeddings reflect an unbiased and comprehensive legal domain representation.
- **Outcome**:
    1. Stakeholders gain confidence in the model's training process, and biases are identified and mitigated early.

Example 2: Attention Weight Analysis

- **Scenario**: Analyzing a model's response to a multi-faceted question.
- **Input Prompt**: "What are the economic impacts of climate change?"
- **Steps**:
    1. Visualize attention distributions in the self-attention and multi-head attention layers.
        - Heatmaps show how tokens like "economic," "impacts," and "climate change" interact.
    2. Examine if attention weights disproportionately focus on irrelevant terms.
        - Example: Excessive focus on "What are" instead of "economic impacts."
    3. Adjust training data or fine-tune the model if necessary.
- **Layer Focus**:
    1. Self-attention and multi-head attention layers: Transparency reveals token relationships and helps refine focus areas.
- **Outcome**:
    1. The analysis ensures the model's reasoning aligns with human expectations, improving trust and usability.

7.KEY FINDINGS

WEEK ONE –

1. Governance frameworks provide structured approaches for mitigating risks but require adaptation for LLM-specific challenges.
2. OWASP risks highlight the importance of protecting sensitive data and securing models against misuse. The research emphasized the necessity for measurable metrics in bias detection, transparency, and explainability.
3. There is a gap in practical guidelines for implementing these governance principles in real-world LLM deployments.
4. Created tables for the 5 government parameters and OSWAP risks for better understanding.

WEEK TWO-

1. Measurement criteria for each parameter were applied to determine their impact on the functionality and ethical alignment of LLM components.
2. The following outcomes were achieved: Identification of LLM Layers:
3. The core layers of LLMs (Tokenization, Embedding, Attention Mechanism, Feed-Forward Networks, Normalization Layers, and Output Layers) were studied to understand their roles in AI operations.
    a. Governance Parameter Mapping: Each governance parameter was mapped to its corresponding LLM layer, highlighting where and how it affects the system's behavior.
    b. Analyising the Measurement Criteria: Specific measurement techniques were decided to evaluate Bias, Explainable AI, Trustworthiness, Transparency, and Privacy at different LLM layers.

WEEK THREE-

1. Bias and fairness are strongly influenced by data representation and tokenization practices.
2. Explainability techniques like SHAP and attention visualization offer actionable insights into LLM decisions.
3. Trustworthiness demands rigorous testing for consistency, factual accuracy, and reliability.
4. Transparency and privacy compliance remain critical for user trust and legal adherence.

WEEK FOUR-

1. Governance challenges manifest differently across layers, requiring tailored interventions.
2. Attention and embedding layers are pivotal for transparency and bias mitigation.
3. Privacy risks are most significant at the embedding and output layers, necessitating robust anonymization and compliance protocols.

8.CONCLUSION

By evaluating the five governance parameters: **Bias, Explainability, Trustworthiness, Transparency, and Privacy,** it identified vulnerabilities and proposed solutions to enhance the ethical alignment of LLMs. Key findings include the importance of attention visualization along with algorithms like SHAP and LIME for explainability, the role of representation metrics in bias mitigation, and the need for differential privacy to protect sensitive data. The resulting governance model and architectural flowchart provide a foundation for designing LLMs that are not only powerful but also responsible, trustworthy, and fair. These outcomes underscore the importance of integrating governance principles into the lifecycle of AI systems to ensure sustainable and ethical innovation.

9.REFERENCES/CITATIONS

[1] Vaswani, A. "Attention is all you need." *Advances in Neural Information Processing Systems* (2017)
[2] Lin, Zichao, Shuyan Guan, Wending Zhang, Huiyan Zhang,Yugang Li, and Huaping Zhang. "Towards trustworthy LLMs: a review on debiasing and dehallucinating in large language models." *Artificial Intelligence Review* 57, no. 9 (2024): 243.
[3] Gallegos, Isabel O., Ryan A. Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K. Ahmed. "Bias and fairness in large language models: A survey." *Computational Linguistics* (2024): 1-79.

[4] Scott, M., and Lee Su-In. "A unified approach to interpreting model predictions." *Advances in neural information processing systems* 30 (2017): 4765-4774.

[5] Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. "" Why should i trust you?" Explaining the predictions of any classifier." In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135-1144. 2016.

[6] Dwork, Cynthia, and Aaron Roth. "The algorithmic foundations of differential privacy." *Foundations and Trends® in Theoretical Computer Science* 9, no. 3–4 (2014): 211-407.

[7] Nasr, Milad, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A. Feder Cooper, Daphne Ippolito, Christopher A. Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. "Scalable extraction of training data from (production) language models." *arXiv preprint arXiv:2311.17035* (2023).
[8] Dai, Wenliang, Tiezheng Yu, Zihan Liu, and Pascale Fung. "Kungfupanda at semeval-2020 task 12: Bert-based multi-task learning for offensive language detection." *arXiv preprint arXiv:2004.13432* (2020).
[9] Glazkova, Anna, Michael Kadantsev, and Maksim Glazkov. "Fine-tuning of pre-trained transformers for hate, offensive, and profane content detection in english and marathi." *arXiv preprint arXiv:2110.12687* (2021).

[10] Brown, Nik Bear. "Enhancing Trust in LLMs: Algorithms for Comparing and Interpreting LLMs." *arXiv preprint arXiv:2406.01943* (2024).

[11] Gebru, Timnit, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé Iii, and Kate Crawford. "Datasheets for datasets." *Communications of the ACM* 64, no. 12 (2021): 86-92.
[12] Bolukbasi, Tolga, Kai-Wei Chang, James Y. Zou, Venkatesh Saligrama, and Adam T. Kalai. "Man is to computer programmer as woman is to homemaker? debiasing word embeddings." *Advances in neural information processing systems* 29 (2016).
[13]Carlini, N., et al. (2023). *Extracting Training Data from Large Language Models*. NeurIPS.
[14]Dwork, C., & Roth, A. (2014). *The Algorithmic Foundations of Differential Privacy*. Foundations and Trends® in Theoretical Computer Science.